Cocoa Tiger Makeover

Session 121

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Cocoa Tiger Makeover

Doug Davidson
Cocoa Frameworks Engineer

http://developer.apple.com/wwdc2005/

Cocoa Tiger Makeover

- Start with basic functionality in a simple application
- Transform into a powerful application using Tiger features
- Goal: you get ideas for “makeover” opportunities in your application

Base Application—iSpend_01

- NSDocument based application
- Master-detail interface using a table view and bindings
- Document is saved in a custom file format

Demo

iSpend_01 Base Application

http://developer.apple.com/wwdc2005/

Makeover Steps

- Pasteboard handling
- User interface improvements
- Searching enhancements

Makeover Steps—iSpend_02

- Pasteboard handling
  - Copy and Paste
  - Drag and Drop
  - Service support
- User interface improvements
- Searching enhancements

Copy and Paste—Writing

Publish list of types:

```
- (NSArray *)writablePasteboardTypes {
   return [NSArray arrayWithObjects:kSpendDocumentType, ...];
}
```

Generic method to write list of types:

```
- (BOOL)writeSelectionToPasteboard:(NSPasteboard *)pboard
        types:(NSArray *)types;
```

Custom code to write specific types:

```
- (BOOL)writeSelectionToPasteboard:(NSPasteboard *)pboard
        type:(NSString *)type {
   if ([type isEqualToString:kSpendDocumentType]) {
      ...
      [pboard setData:data forType:kSpendDocumentType];
   } else ...
}
```

Copy and Paste—Reading

Publish list of types:

```
- (NSArray *)readablePasteboardTypes {
   return [NSArray arrayWithObjects:kSpendDocumentType, ...];
}
```

Generic method to read list of types:

```
- (BOOL)readSelectionFromPasteboard:(NSPasteboard *)pboard;
```

Custom code to read specific types:

```
- (BOOL)readSelectionFromPasteboard:(NSPasteboard *)pboard
        type:(NSString *)type {
   if ([type isEqualToString:kSpendDocumentType]) {
      NSData *data = [pboard dataForType:kSpendDocumentType];
      ...
   } else ...
}
```

Demo

iSpend_02 Copy and Paste

http://developer.apple.com/wwdc2005/

Drag and Drop—Source

Use NSTableView data source support—improved for Tiger!

Set source operation mask:

```
[tableView setDraggingSourceOperationMask:mask forLocal:flag];
```

Write to pasteboard at start of drag:

```
- (BOOL)tableView:(NSTableView *)tv
     writeRowsWithIndexes:(NSIndexSet *)rowIndexes
     toPasteboard:(NSPasteboard *)pboard {
   ...
   [document writeSelectionToPasteboard:pboard
             types:[document writablePasteboardTypes]];
```

Handle promise drags:

```
- (NSArray *)tableView:(NSTableView *)tv
   namesOfPromisedFilesDroppedAtDestination:(NSURL *)dest
   forDraggedRowsWithIndexes:(NSIndexSet *)rowIndexes {
   ...
   [document writeSelectionToDestination:dest];
```

Drag and Drop—Destination

Use NSTableView data source support—improved for Tiger!

Register for types:

```
[tableView registerForDraggedTypes:[document readablePasteboardTypes]];
```

Validate drags as they arrive:

```
- (NSDragOperation)tableView:(NSTableView *)tv
   validateDrop:(id <NSDraggingInfo>)info proposedRow:(int)row
   proposedDropOperation:(NSTableViewDropOperation)op;
```

Accept drops:

```
- (BOOL)tableView:(NSTableView *)tv
   acceptDrop:(id <NSDraggingInfo>)info row:(int)row
   dropOperation:(NSTableViewDropOperation)op {
    return
      [document readSelectionFromPasteboard:[info draggingPasteboard]];
}
```

Services—Using Services

Register types:

```
[NSApp registerServicesMenuSendTypes:[self writablePasteboardTypes]
       returnTypes:[self readablePasteboardTypes]];
```

Validate services menu entries:

```
- (id)validRequestorForSendType:(NSString *)sendType
                     returnType:(NSString *)returnType;
```

Handle pasteboard:

```
- (BOOL)writeSelectionToPasteboard:(NSPasteboard *)pboard
        types:(NSArray *)types;
- (BOOL)readSelectionFromPasteboard:(NSPasteboard *)pboard;
```

Services—Providing Services

Advertise in Info.plist:

```
<key>NSServices</key>
<array>
    <dict>
        <key>NSMessage</key>
        <string>importData</string>
        ...
```

Set service provider:

```
[NSApp setServicesProvider:provider];
```

Handle service requests:

```
- (void)importData:(NSPasteboard *)pboard
          userData:(NSString *)data
          error:(NSString **)error {
    ...
    [document readSelectionFromPasteboard:pboard];
}
```

Demo

iSpend_02 Drag & Drop and Services

http://developer.apple.com/wwdc2005/

Cocoa Tiger Makeover

User Interface Improvements

Hansen Hsu
Cocoa Frameworks Engineer

http://developer.apple.com/wwdc2005/

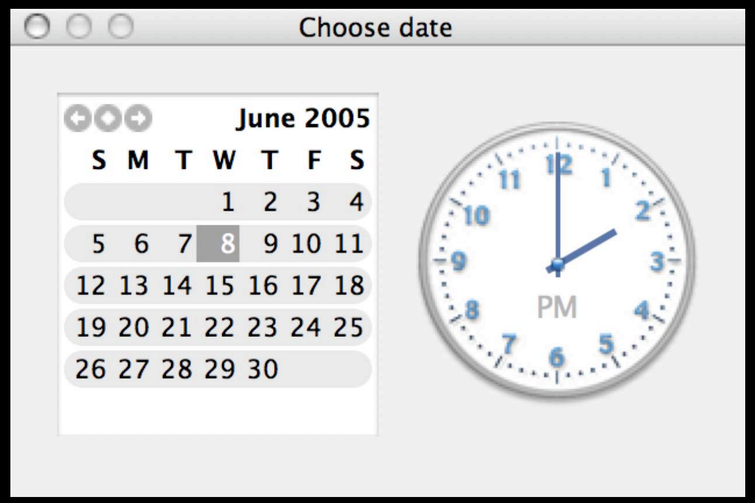
Makeover Steps—iSpend_03

- Pasteboard handling
- User interface improvements
  - Date picker
  - Undo
  - View animation

Date Picker

- New in Tiger!
- Allows you to display an NSDate instance
- Textual NSDatePicker provides a text field and stepper

- Graphical NSDatePicker provides a calendar and clock



- You specify which date and time elements should be visible

Demo

iSpend_03 Date Picker

http://developer.apple.com/wwdc2005/

Undo Support

- NSUndoManager records operations for undo and redo
- Register with undo manager to indicate how to revert a change
- NSDocument instance supplies an undo manager
- The undo manager also updates edited state of document
- CoreData provides undo support for free

Types of Undo Registration

- Simple undo registration
  - Use when change can be undone with method taking single object
  - Specify target, method, and argument

```
[undoManager registerUndoWithTarget:self
             selector:@selector(setDate:)
             object:_date];
```

- Invocation-based undo registration
  - Use when there are multiple arguments, or argument is not an object
  - Specify actual message to revert object state

```
[[undoManager prepareWithInvocationTarget:self]
    insertObjects:oldTransactions
    inTransactionsAtIndexes:oldIndexes];
```

Adding Undo Support

- Undo addition or removal of transactions in document
  - Use Key Value Observing to observe changes to transactions array
- Undo change to Transaction value
  - Retrieve undo manager from document instance
  - Register undo instructions in each setter method

Demo

iSpend_03 Undo

http://developer.apple.com/wwdc2005/

View Animation

- New in Tiger!
- NSAnimation manages timing and progress of animations
- NSViewAnimation subclass provides simple visual transitions
  - Applies to a view or window
  - Fade in or out
  - Resize or move

View Animation

- NSViewAnimation can manage multiple animations at once

```
[[NSViewAnimation alloc] initWithViewAnimations:viewAnimations];
```

- Each viewAnimation is specified with a dictionary

```
[NSDictionary dictionaryWithObjectsAndKeys:
    newView,                        NSViewAnimationTargetKey,
    NSViewAnimationFadeInEffect,    NSViewAnimationEffectKey,
    startFrame,                     NSViewAnimationStartFrameKey,
    endFrame,                       NSViewAnimationEndFrameKey,
    nil]
```

Related session

Cocoa Advanced View Techniques

Session 146
Friday 3:30 PM

Using View Animation

- Create three view animation dictionaries
  - Fade out and position old view
  - Resize the document window
  - Fade in and expand new view
- Create NSViewAnimation with list of animation dictionaries
- Run the animation
- When animation finishes, update state

Demo

iSpend_03 View Animation

http://developer.apple.com/wwdc2005/

Cocoa Tiger Makeover

Searching Enhancements

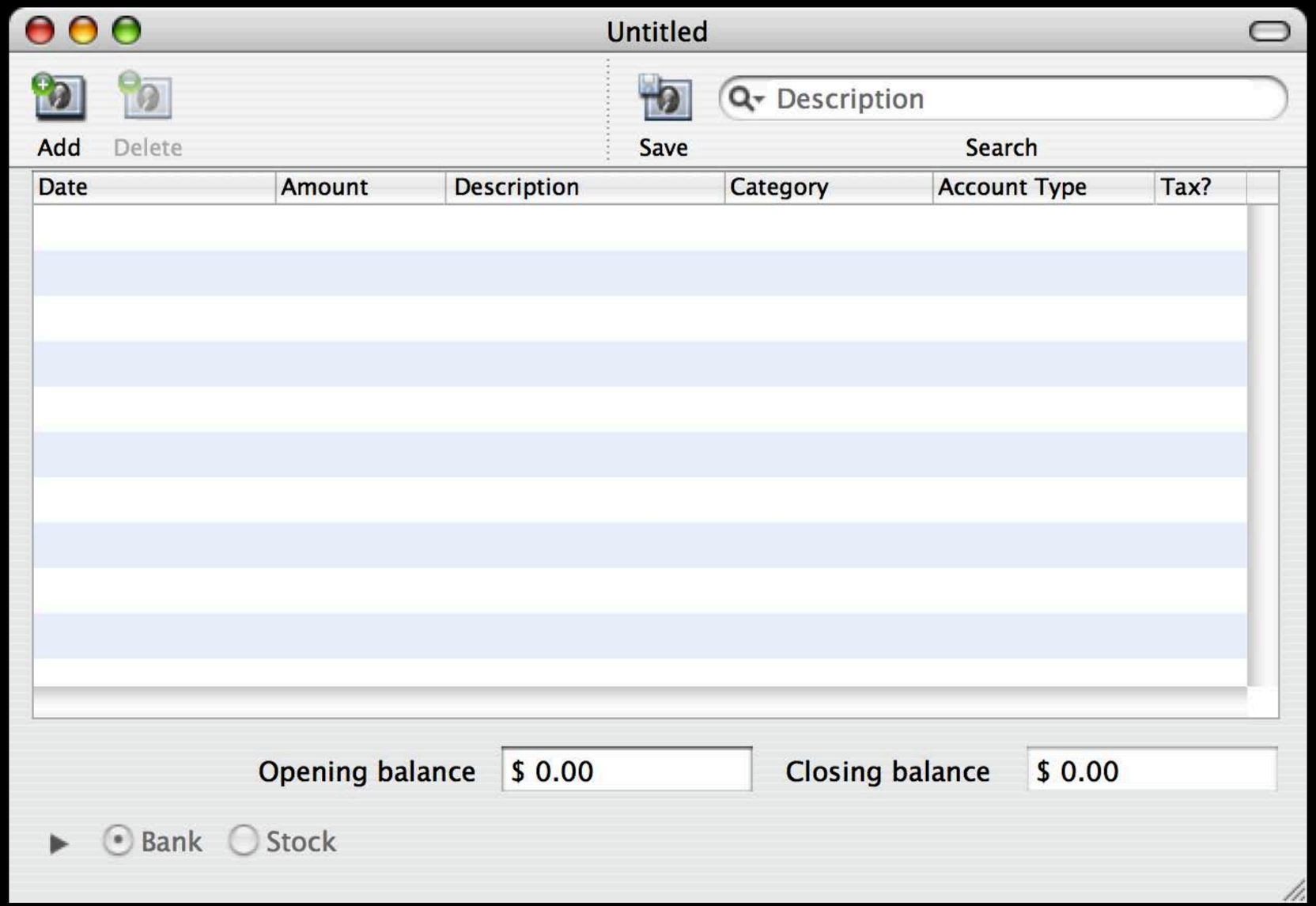
Corbin Dunn
Cocoa Frameworks Engineer

http://developer.apple.com/wwdc2005/

Makeover Steps—iSpend_04, iSpendPlugin

- Pasteboard handling
- User interface improvements
- Searching enhancements
  - Toolbar Search Field
  - Spotlight Metadata Plugin
  - Spotlight Search Window

Toolbar Search Field



Toolbar Search Field

- Add an NSCustomView to the MyDocument nib file
- Add an NSSearchField to the view
- Add an outlet: NSSearchField *searchFieldOutlet;
- NSSearchField bindings—*new in Tiger!*
  - Bind predicate with “descriptionString contains[c] $value”
    - Name:“Description”
  - Bind predicate2 with “type contains[c] $value”
    - Name:“Category”
  - More NSPredicate information later!
- Add some code to create the toolbar item

Demo

iSpend_04 Toolbar Search Field

http://developer.apple.com/wwdc2005/

Spotlight Metadata Plug-in

- Allows the Spotlight database to “know” about your file types

```
UTI - com.apple.ispend.document
```

Spotlight Metadata Plug-in

- Allows the Spotlight database to “know” about your file types

```
UTI - com.apple.ispend.document
```

- Allows us to fill in standard attributes:

```
kMDItemTextContent—Text content—see MDItem.h
```

Spotlight Metadata Plug-in

- Allows the Spotlight database to "know" about your file types

```
UTI - com.apple.ispend.document
```

- Allows us to fill in standard attributes:

```
kMDItemTextContent–Text content–see MDItem.h
```

- Allows custom attributes for your types:

```
com_apple_ispendBalance–Total balance
```

The iSpend Metadata Plug-in

- A new “Metadata Importer” Plug-in created in Xcode
- Proper entries in schema.xml and Info.plist added

Fill in GetMetadataForFile.m

- Add code to GetMetadataForFile.m, implement:

```
Boolean GetMetadataForFile(void* thisInterface,
          CFMutableDictionaryRef attributes,
          CFStringRef contentTypeUTI,
          CFStringRef pathToFile)
```

Fill in GetMetadataForFile.m

- Add code to GetMetadataForFile.m, implement:

```
Boolean GetMetadataForFile(void* thisInterface,
           CFMutableDictionaryRef attributes,
           CFStringRef contentTypeUTI,
           CFStringRef pathToFile)
```

- Conceptually what will be done:
  - Load `pathToFile` into an NSData object and deserialize it:

```
NSDictionary *documentDictionary =
   [NSPropertyListSerialization
      propertyListFromData:data
      mutabilityOption:NSPropertyListImmutable
      format:NULL
      errorDescription:&errorString];
```

Fill in GetMetadataForFile.m

- Add code to GetMetadataForFile.m, implement:

```
Boolean GetMetadataForFile(void* thisInterface,
           CFMutableDictionaryRef attributes,
           CFStringRef contentTypeUTI,
           CFStringRef pathToFile)
```

- Conceptually what will be done:
  - Load `pathToFile` into an NSData object and deserialize it:

```
NSDictionary *documentDictionary =
   [NSPropertyListSerialization
      propertyListFromData:data
      mutabilityOption:NSPropertyListImmutable
      format:NULL
      errorDescription:&errorString];
```

- Gather the combined descriptions, balances

Compile and Install

- Compile the project in Xcode
- Install in: ~/Library/Spotlight, /Library/Spotlight, or /System/Library/Spotlight

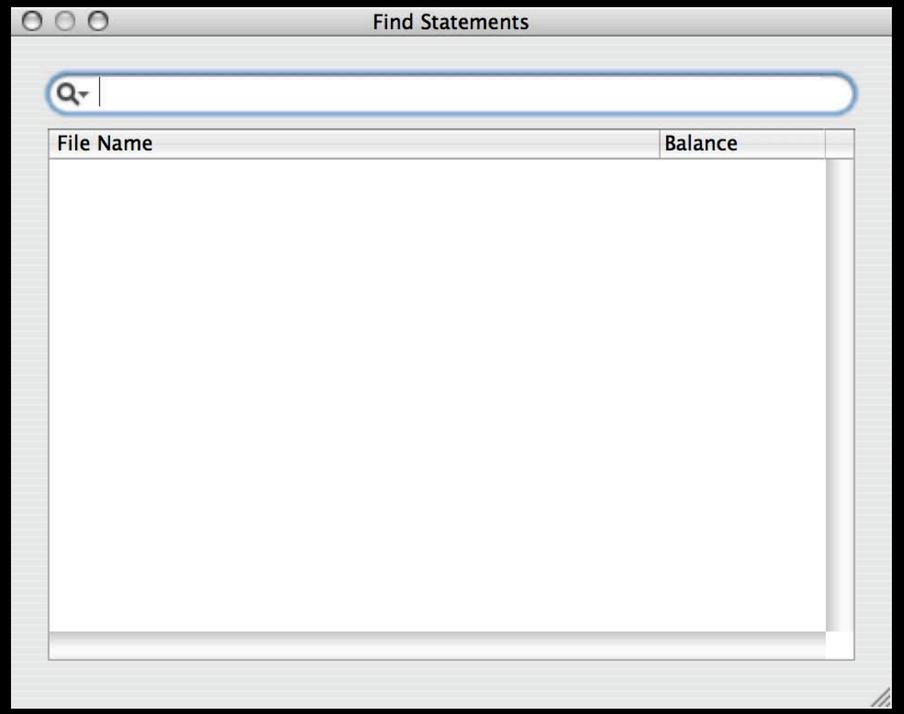
Demo

iSpendPlugin Spotlight Metadata Plug-in

http://developer.apple.com/wwdc2005/

Spotlight Search Window

- Search the system for all documents, or just your documents
- What is already done: a basic window, shown from the menu:
  - View -> Search Window
- Created a nib with a window named SearchPanel and added:
  - NSSearchField
  - NSTableView—with two columns
- Created a controller named SearchController.h/.m



Editing the SearchController.h

- NSMetadataQuery—the Cocoa entry point into searches

```
@interface SearchController : NSObject {
    NSMetadataQuery *_query;
    NSString *_searchKey;
   ...
}
```

- _searchKey will be used to bind the search key value in the NSSearchField
- _query will also be exposed for bindings

Exposing Things for Bindings

- Expose the query for bindings:

```
- (NSMetadataQuery *)query {
    return _query;
}
```

- Allowing us to bind to:

```
@interface NSMetadataQuery : NSObject
...
- (NSArray *)results;
...
```

- Which is an array of NSMetadataItems

Exposing Things for Bindings

- Expose the searchKey:

```
- (NSString *)searchKey {
    return [[_searchKey copy] autorelease];
}

- (void)setSearchKey:(NSString *) value {
    if (_searchKey != value) {
        [_searchKey release];
        _searchKey = [value copy];
        if (_searchKey == nil) _searchKey = @"";
        [self createSearchPredicate];
    }
}
```

NSPredicate—What Is It?

- Used to define logical conditions for searching or filtering
- For example, “description LIKE ‘Tom’”
- NSCompoundPredicate allows combining predicates (AND, OR, NOT) to create a new predicate based on two expressions
- Read the documentation for NSPredicate for more information

Create the Search Predicate

- (void)createSearchPredicate contains:

```
NSPredicate *predicateToRun = [NSPredicate
  predicateWithFormat:@"kMDItemTextContent like[c] %@",
  _searchKey];

NSPredicate *ourTypesPredicate = [NSPredicate
  predicateWithFormat:
  @"kMDItemContentType == 'com.apple.ispend.document'"];

predicateToRun = [NSCompoundPredicate
  andPredicateWithSubpredicates:
     [NSArray arrayWithObjects:
        predicateToRun, ourTypesPredicate, nil]];

[_query setPredicate:predicateToRun];

[_query startQuery];
```

Set the Bindings in the UI

- Add an NSArrayController—bind contentArray to query.results
- TableColumn 1—“File Name”
  - Bind value to kMDItemFSName in arrangedObjects
- TableColumn 2—“Balance”
  - Bind value to com_apple_ispendBalance in arrangedObjects

Demo

iSpend_04 Spotlight Search Window

http://developer.apple.com/wwdc2005/

Makeover Complete

- Pasteboard handling
- User interface improvements
- Searching enhancements

Cocoa “Tiger Makeover” Lab | **Wednesday 6:30PM**

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Who to Contact

Matthew Formica
Cocoa / Core Data Evangelist
mformica@apple.com

Cocoa Development
Apple Email Discussion List
cocoa-dev@lists.apple.com

Bug Reporter
Apple Bug Tracking System
http://bugreport.apple.com

WWDC2005

Worldwide Developers Conference